A.
Maze Parser

For this lab, you will be writing a program to download a web page that generates a maze, then parse through it, and load it up into a Maze class. To understand this better, take a look at the maze generator website: http://www.delorie.com/game-room/mazes/genmaze.cgi . Set “type” to “text” and leave “seed” blank. Try a few different dimensions, and play around with the width / height.

Now, the code to download the maze is already provided for you in mazeDownloader.cpp and mazeDownloader.h. You just need to look at the .h file, and include it at the top of your own file: #include “mazeDownloader.h”. Your job is to make the main.cpp provided to you work. This means you need to make a Maze class with a with the following functions:

1. Constructor + deconstructor
2. Download()
3. ParseMazeOut()
4. Overloaded input and output operators

The input operator should allow the user to take in the following pieces of information: download file name, rows, columns, width, and height. You should then be able to call Download and it will save to filename you give it. Now, if you examine the downloaded file, you see it has a bunch of HTML surrounding the actual maze. Observe carefully the tags (html tags are words surrounded by < > brackets) near the maze. Are any of them unique enough to identify where the maze starts or ends in the file? You can’t just rely on the maze being always on the ___ line of the file. You need to copy the maze into a 2D char array. Once you do that, you should be able to overload the output operator so that it can be displayed easily. If you are interested in how downloading the web page works, read the optional last paragraph. Here is an example of how the program should run:

Important note: Look at how it was compiled. Main and mazeDownloader are already provided. You just list all your .cpp files after g++, and at the end tell g++ where to locate the libcurl library. Including libraries is done by using a **-l**. That is a **lower case L**, not a 1 or a capital i. Just add -l curl at the end.

(Optional, if you're interested) About the web page downloading: It's a little clunky to do in C++, but libcURL is a nice library for accessing web pages in C/C++. Open up the mazeDownloader.cpp file provided and take a look! It might seem confusing because it's written in C, not C++, but hopefully you get the idea. A function is also provided for you to just give a URL and have it download the corresponding web page. If you want to do more, here is a tutorial: https://curl.haxx.se/libcurl/c/libcurl-tutorial.html . And here is a bunch of examples / documentation: https://curl.haxx.se/libcurl/c/example.html. I highly encourage you to make a little program on your own using this. One student made a program to download a humane society page every 10 minutes, and email him if a Husky showed up!

B.
Solve the maze

Now we want to add a character that you can move through the maze. A file, main2.cpp is already provided for you, and you should not modify it. This file has what is typically called a "Game loop" in it. Game loops usually follow along the lines of this: display the current state of things, wait for the user to give some input, update the state of things according to what the user did, and repeat. In this case, we are displaying the board and character, waiting to see how the character is moved, moving the character if that move was valid at all, and repeating unless the character hits the exit. You should move the character through the w,a,s,d keys. If the character

tries to go off the map or into a wall, do not move the character. Make a character class, and give it whatever member variables / functions you think it needs.
Two iterations are shown below as an example: